

REMARKS

This Amendment is responsive to the Office Action mailed February, 21, 2007.

In the Office Action, claims 2, 20 and 38 were rejected as being indefinite under 35 U.S.C. §112(2). Reconsideration and withdrawal of these rejections are respectfully requested.

Claims 2, 20 and 38 recite: “assembling the requested document from at least one of the retrieved and dynamically generated blocks.” This is the standard, well understood “at least one of A and B” form of claim construction, which is equivalent to “A and/or B.” Accordingly, “at least one of A and B” includes, within its scope:

- A only;
- B only, or
- Both A and B

Therefore, the recitation “assembling the requested document from at least one of the retrieved and dynamically generated blocks” includes, within its scope:

- assembling the requested document from the retrieved blocks;
- assembling the requested document from the dynamically generated blocks, or
- assembling the requested document from the retrieved blocks *and* from the dynamically generated blocks

It is respectfully submitted, therefore, that claims 2, 20 and 38 define steps with the requisite degree of particularity for the purposes of 35 USC §112(2).

It is also respectfully submitted that the language of claims 2, 20 and 38 is not exemplary in nature. Therefore, MPEP 2173.05(d) is inapplicable here. MPEP 2173.05(d) is reproduced herein below:

2173.05(d) Exemplary Claim Language (“for example,” “such as”) [R-1] - 2100 Patentability

2173.05(d) Exemplary Claim Language (“for example,” “such as”) [R-1]

Description of examples or preferences is properly set forth in the specification rather than the claims. If stated in the claims, examples and preferences >may< lead to confusion over the intended scope of a claim. In those instances where it is not clear whether the claimed narrower range is a limitation, a

rejection under 35 U.S.C. 112, second paragraph should be made. The examiner should analyze whether the metes and bounds of the claim are clearly set forth. Examples of claim language which have been held to be indefinite because the intended scope of the claim was unclear are:

- (A) "R is halogen, for example, chlorine";
- (B) "material such as rock wool or asbestos" *Ex parte Hall*, 83 USPQ 38 (Bd. App. 1949);
- (C) "lighter hydrocarbons, such, for example, as the vapors or gas produced" *Ex parte Hasche*, 86 USPQ 481 (Bd. App. 1949); and
- (D) "normal operating conditions such as while in the container of a proportioner" *Ex parte Steigerwald*, 131 USPQ 74 (Bd. App. 1961).

The above examples of claim language which have been held to be indefinite are fact specific and should not be applied as *per se* rules. See MPEP § 2173.02 for guidance regarding when it is appropriate to make a rejection under 35 U.S.C. 112, second paragraph.

The claim language "assembling the requested document from at least one of the retrieved and dynamically generated blocks" is not exemplary in nature, but requires one of the following:

- assembling the requested document from the retrieved blocks;
- assembling the requested document from the dynamically generated blocks, or
- assembling the requested document from the retrieved blocks *and* from the dynamically generated blocks.

Indeed, the claim does not include any exemplary language ("such as", "for example", etc.), which would trigger MPEP 2173.05(d). Reconsideration and withdrawal of these rejections are respectfully requested.

Claims 1-60 were rejected under 35 U.S.C. §101, as allegedly not defining patentable subject matter. Reconsideration and withdrawal of these rejections are respectfully requested.

As the Examiner will note, claims 1-60 have been amended in exactly the manner helpfully suggested by the Examiner to overcome these §101 rejections. Amended claims 1-60 are believed, therefore, to define subject matter that squarely fits within the purview of 35 U.S.C. §101. Reconsideration and withdrawal of these rejections are respectfully requested.

Claims 1-54 were rejected under 35 U.S.C. §103(a) as being unpatentable over a five-way combination of McLaughlin et al., Nazem et al., Michel K. Bowman-Amuah, Robert S.

Mason and Bender et al.. Reconsideration and withdrawal of these rejections are respectfully requested.

At the outset, claim 1 recites:

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document; and

dynamically generating remaining ones of the plurality of blocks defined in the script of the requested document that are not stored in the memory and storing a copy of each dynamically generated block in the memory.

The claim requires that only some of the blocks be retrieved from the memory and that remaining ones of the plurality of blocks be dynamically generated. The claimed embodiment, therefore, requires two actions: 1) retrieving only some of the blocks from the memory; and 2) dynamically generating remaining ones of the plurality of blocks. The applied combination does not teach or suggest such a method.

The applied combination, however, does not teach or suggest that only some of the blocks be retrieved from a memory and that all remaining blocks defined in the script be dynamically generated, as claimed herein. Indeed, the primary reference to McLaughlin et al. teaches an all or nothing approach to serving a client application with process data:

When a client application requires process data, a data request is sent to supervisory controller 120 and is received in cache manager 220. In one embodiment of the present invention, cache manager 220 first searches dynamic cache 220 for the requested process data. If the process data is found (a cache hit), cache manager 220 transfers the process data to the requesting client application and the transaction is ended. If the data is not found (a cache miss), cache manager 220 then requests the process data from the appropriate one of process nodes 204–206. When the supervisory controller 120 receives the process data from the process node, cache manager 215 writes the process data into dynamic cache 215 and transfers the process data to the client application that originally requested it.

as shown in Col. 8, lines 6-19. Either process data is present in the cache 220 or it is fetched from one of the process nodes 204-206. No provisions are made within McLaughlin et al. for

retrieving only some of the blocks from the memory and dynamically generating remaining ones of the plurality of blocks. Alternatively, McLaughlin et al. teach that process data may be obtained in a peer-to-peer fashion, as detailed in Col. 6, lines 22-25 or via a publish/subscribe model, as detailed in Col. 3, lines 55-57.

Similarly, all of the supervisory data is dynamically generated, as positively stated in Col. 5, lines 33-36:

optimize the facility as a whole. In a preferred embodiment, the supervisory data is dynamically generated and is based at least upon a given facility's efficiency, production or economic cost, and most preferably all three. 35

i.e., none of the supervisory data is retrieved from a cache memory. Therefore, McLaughlin et al. do not teach or suggest retrieving only some of the blocks from the memory and dynamically generating remaining ones of the plurality of blocks, as required by the claims.

The outstanding Office Action states that Nazem et al. teach a document that includes a plurality of blocks. However, Nazem et al. do not remedy the fundamental shortcomings of the primary reference and do not teach – alone or in combination with McLaughlin et al. - to retrieve only some of the blocks from the memory and to dynamically generate remaining ones of the plurality of blocks, as claimed herein. Indeed, Nazem et al. teach the use of templates to hold the static data and to obtain all live data from the shared memory 212:

in further detail. User front page 218 is built according to a user template and live data. The user template specifies, for example which quotes are shown in the portfolio module, which cities are displayed in the weather module, etc. Each of the modules 504 can be customized by a user and moved about front page 218. The modules 504 are also reusable, in that any customized module which appears on multiple pages can be edited from any one of those pages and the edits will be reflected on each of the pages. Other custom pages for the user can be viewed by selecting one of the page buttons 502 appearing below the header. Other pages and utilities can be selected using the buttons 508 which are part of the header. 55 60 65

In addition to all of the live data shown in FIG. 5 being stored in the shared memory, summaries from each of the

as stated in line 66-67 above. Nazem et al. teach that the template data is stored (and necessarily retrieved from) the cached user templates database 214 and that all of the live data to fill these templates is stored in (and necessarily retrieved from) the shared memory 212. Therefore, even when considered collectively with the McLaughlin et al. reference, Nazem et al. do not teach or suggest to retrieve only some of the blocks from the memory and to dynamically generate remaining ones of the plurality of blocks, as required by each of the independent claims herein.

It is respectfully submitted that adding the Bowman-Amuah reference to the mix does not provide any additional guidance to the person of ordinary skill in this art. The cited passages simply do not teach anything relevant to blocks of a document including a reference to a data source and code that is configured to access and format data accessed from the data sources, as required by the claims. Indeed, Col. 52, lines 55-61 deals with ID and password pairs:

55 resources, as opposed to securing an applications detailed functions.

The security component prevents unauthorized users from accessing corporate data/resources by providing the users with access codes—password & ID—that allows the user to 60 login to the system or execute any (or a particular) application.

while Col. 47, lines 30-67 deals with wholly unrelated performance issues:

How important is performance?

In general, performance of data access and printing should be considered. Some typical benchmark tests include table scan, single-table report, joined table report, and mailing label generation times. (source is market research)

What is the budget?

Per developer costs as well as run time licensing fees, maintenance costs, support fees, and upgrade charges should be considered.

Do I have another component that satisfies this requirement?

Many databases and application development tools are shipped with built-in or add-on report writing capability. However, stand-alone report writers: (1) are more powerful and flexible, especially when dealing with multiple data sources and a wide variety of formats; (2) can retrieve information from more data sources than the bundled report writers and can create reports from several data sources simultaneously; (3) excel in ease of use, both in designing and generating reports; (4) offer better tools and more predefined reports; and (5) have faster engines. (source is market research)

Does the product integrate with the existing or proposed architecture?

It is important to consider how well a product integrates with desktop tools (word processing, spreadsheet, graphics etc.) and application development programs. These items can be used to extend the capabilities of the reporting package.

What databases does the product support?

A product should support the most widely used PC file formats and Client/Server databases. It may be necessary to consider the type of support. For example, native database interfaces tend to have better performance than open standards such as ODBC. Another possible consideration is how well the product accesses multiple files or databases. (source is market research)

Next, the Office turns to Mason, Jr. for a supposed teaching of “retrieving only some of the plurality of blocks, and generating remaining blocks,” and points to Col. 8, lines 1-28 and Col. 9, lines 1-30 in support of its §103(a) rejection. However, Mason, Jr., whether considered alone or in combination with McLaughlin et al., Nazem et al. (5,983,227) and/or Bowman-Amuah does not teach or suggest the claimed subject matter.

Indeed, Mason, Jr. teaches a RAID controller system, and not a method of servicing a request for a document over a computer network, as claimed. Mason, Jr. teaches, at Col. 8, lines 1-30 referred to by the Examiner, that in case of a read operation failure, the data is reconstructed using a corresponding parity block of the stripe in which the failure occurred. The back end cache must then determine whether the parity block has already been cached by referring to a back end cache block list. If it is, then a read I/O operation may be avoided. Any remaining blocks of the stripe are read and cached in a front end cache. Then all blocks of the stripe in which the failure occurred are XORed together with the parity block, whereupon valid data is passed to the front end cache. No teaching or suggestion are present in this passage of Mason, Jr., whether considered alone or in combination with the other references, that would lead a person of ordinary skill in the art to the claimed embodiments – that is, to retrieve only some of the blocks from the memory and to dynamically generate remaining ones of the plurality of blocks .

Next, the Office points to Col. 9, lines 1-30 of Mason, Jr. This passage teaches a write command in which a list of valid blocks is checked. The list (see Col. 8, lines 56-61) accompanies a write command received from a host, includes the new blocks to be written to the RAID array. Mason, Jr. teaches, in Col. 9, beginning at line 4, teaches that if some old blocks to be written are missing from the list (meaning that a cache miss has occurred), they are retrieved

from the physical disk and stored into the front end cache. In essence, Mason, Jr. teaches that if blocks to be written are missing from the list, they are retrieved from the physical disks. At the outset, kindly note that this reference only teaches operations carried out within a RAID array of hard disk drives, and does not teach servicing a request for a document over any computer network. Moreover, note that Mason, Jr. teaches that if blocks to be written are missing from the list, they are **retrieved** from the physical disks, which cannot be considered to teach or to suggest dynamically **generating** remaining ones of the plurality of blocks ... that are not stored in the memory. When blocks are missing from the list of blocks to be written, Mason, Jr. teaches simply to retrieve them from the disk. The claimed embodiment, on the other hand, requires dynamically generating remaining ones of the plurality of blocks, which is unsuggested by Mason, Jr., whether considered alone or in combination with the other constituent references of the applied combination. Moreover, it is respectfully submitted that the skilled artisan would not look to or find any guidance in a patent relating to a RAID controller when seeking to develop a method for servicing requests for documents over a computer network, as is the Mason, Jr. patent.

As the four-way McLaughlin et al.-Nazem et al.-Bowman-Amuah-Mason, Jr. combination was **overcome** by the Amendment filed November 4, 2006. The Office added Bender et al. to the applied combination of references. In the outstanding Office Action, the Office states “McHaghlín, Nazeem and Bowman-Amuah do not explicitly disclose retrieving only some of the plurality of blocks,” but states that Bender et al. “discloses retrieving only some of the plurality of blocks [see Col. 18, lines 60-65]” and also points to Bender et al.’s Col. 4, lines 12-14 in support of the assertion that Bender et al. teach retrieving only some of the plurality of blocks.

It is respectfully submitted, however, that Bender, et al. teach nothing of the sort. The very passage cited by the Examiner does not even support the Examiner's own position. Indeed, Col. 18, lines 60-65 states:

60 (a) successively retrieving each of said plurality of blocks
of uncompressed image data in said memory circuit and
communicating said blocks to said image output
device, and increasing said Request Counter's value
65 upon each occurrence of retrieving one of said plurality
of blocks; and

This passage clearly states that EACH of the plurality of blocks is successively retrieved. It does not teach or suggest, whether considered alone or in combination with McLaughlin et al.-Nazem et al.-Bowman-Amuah-Mason, Jr. the subject matter of the independent claims; namely:

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document;

Indeed, the claim clearly requires that only some of the plurality of blocks be retrieved. However, Bender, et al. teach that EACH of the plurality of blocks are retrieved. Therefore, far from teaching the claimed embodiment (whether alone or in combination with the other references), Bender et al. teach away from the claimed embodiment – by teaching that EACH of the plurality of blocks is retrieved, as opposed to “only some”, as claimed in each of the pending independent claims.

It is respectfully submitted that the motivation ascribed to the person of ordinary skill in the art, gleaned from Bender et al., at Col. 4, lines 12-14:

It is a further object of the present invention to provide a printer that both operates more efficiently and with less memory by operating in parallel by continuing to process

is wholly unrelated to the claims. Therefore, even if a person of ordinary skill in the art were motivated by a desire to (provide a printer that operated) [operate] more efficiently ... by

operating in parallel,” (for some unknown reason) and retrieved EACH of the plurality of blocks as taught by Bender et al., the claimed embodiments would not result.

It is respectfully submitted that the applied combination, therefore, does not teach or suggest any method, system or media configured to:

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document; and

dynamically generating remaining ones of the plurality of blocks defined in the script of the requested document that are not stored in the memory and storing a copy of each dynamically generated block in the memory.

as claimed in independent claims 1, 19 and 37. The “only some” language clearly precludes any interpretation wherein EACH of the plurality of blocks are retrieved, as Bender et al. would have it.

Reconsideration and withdrawal of the outstanding art rejections based upon the five-way combination of McLaughlin et al., Nazem et al., Bowman-Amuah, Mason, Jr. and Bender et al. are respectfully requested.

Claims 55 and 57-60 were rejected as being unpatentable over a four-way combination of McLaughlin et al., Nazem et al. (Mason et al., is applied on page 12, although not listed as an applied reference in paragraph 34) and Bender et al. Reconsideration and withdrawal of these rejections are respectfully requested.

As the Office will note, claim 55 has been amended to recite:

**servicing the request for the Web page by assembling the requested Web page by retrieving only some of the cached blocks of the requested Web page from the memory and generating all remaining blocks of the requested Web page that were not retrieved from the memory memory, and
sending the assembled Web page over the computer network.**

for the purpose of addressing the §101 rejection only. Note the “retrieving only some of the cached blocks of the requested Web page from the memory” claim language in claim 55. It is

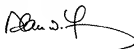
respectfully submitted that all of the above comments relative to McLaughlin et al., Nazem et al., Mason et al. and Bender et al. are also applicable to claim 55. Rather than repeat these arguments here, they are hereby incorporated herein by reference as if they had been repeated here in full.

Bender et al.: a) simply do not teach "retrieving only some of the cached blocks of the requested Web page from the memory," as claimed; b) do not remedy the shortcomings of the primary combination to McLaughlin et al. and Nazem et al.; and c) is not, as stated, "in the same field of endeavor" as Applicants (compressing and decompressing print data in the background of a printer has little or nothing to do with servicing a request for a Web page over a computer network, as claimed herein). Reconsideration and withdrawal of these rejections are respectfully requested.

As the rejections of the independent claims are deemed to have been overcome, it is not believed necessary to discuss the rejections of the dependent claims at this time, as they incorporate the patentable features of the independent claims from which they depend.

Applicants believe that this application is now in condition for allowance. If any unresolved issues remain, please contact the undersigned attorney of record at the telephone number indicated below and whatever is necessary to resolve such issues will be done at once.

Respectfully submitted,



Date: May 21, 2007

By: _____

Alan W. Young
Attorney for Applicants
Registration No. 37,970

YOUNG LAW FIRM, P.C.
4370 Alpine Rd., Ste. 106
Portola Valley, CA 94028
Tel.: (650) 851-7210
Fax: (650) 851-7232

\\Ylfserver\y\l\CLIENTS\ORCL\5680\5680 AMEND.4.docx